(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 June 2002 (27.06.2002)

PCT

(10) International Publication Number
WO 02/50678 A1

(51) International Patent Classification[7]: G06F 11/00, 11/20, 11/30, 11/14, 11/16

(21) International Application Number: PCT/US01/49600

(22) International Filing Date:
19 December 2001 (19.12.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/257,478 21 December 2000 (21.12.2000) US
09/855,592 14 May 2001 (14.05.2001) US

(71) Applicant: **LEGATO SYSTEMS, INC.** [US/US]; 2350 West El Camino Real, Mountain View, CA 94040 (US).

(72) Inventor: **PRICE, Daniel, M.**; 11030 Manor Circle, Highland, UT 84003 (US).

(74) Agents: **ISRAELSEN, R., Burns** et al.; Workman, Nydegger & Seeley, 1000 Eagle Gate Tower, 60 East South Temple, Salt Lake City, UT 84111 (US).

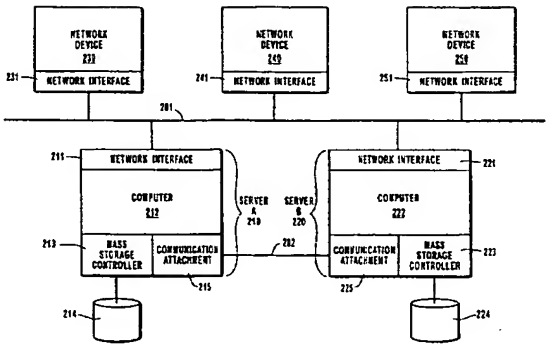
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: A METHOD OF IMPROVING THE AVAILABILITY OF A COMPUTER CLUSTERING SYSTEM THROUGH THE USE OF A NETWORK MEDIUM LINK STATE FUNCTION



(57) Abstract: A method for increasing the availability of a first server (210) included in a computer cluster when a second server (220) fails. Each server (210, 220) in the computer cluster has an associated mass storage device 214, 224 and can process requests from any network device (230, 240, 250) in the computer cluster. Data is mirrored between the mass storage devices (214, 224) of the servers (210, 220) so that each server's mass storage device has a complete copy of all computer cluster data. Data mirroring takes place across a dedicated link (202), which reduces congestion on the rest of the computer cluster. When the first server (210) detects a loss of communication from the second server (220), the first server (210) determines if the loss of communication is a result of a malfunction of the dedicated link (202). If the dedicated link (202) has failed, the first server (210) discontinues operation to avoid writing data to its associated mass storage device (214), which cannot be mirrored due to the loss of communication. If the dedicated link (202) is operational, the first server (210) continues operation. In either case, since each server (210, 220) can process requests from any network device (230, 240, 250) and each server has a complete copy of all the network data, the computer cluster continues to be available for use even after a server is shut down.

WO 02/50678 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A METHOD OF IMPROVING THE AVAILABILITY OF A COMPUTER CLUSTERING SYSTEM THROUGH THE USE OF A NETWORK MEDIUM LINK STATE FUNCTION

BACKGROUND OF THE INVENTION

1. The Field of the Invention

This invention relates to computer clustering systems and in particular to methods for improving the availability and reliability of computer clustering system resources and data in the event of loss of communication between computer clustering system servers.

2. Description of Related Art

A typical computer cluster includes two or more servers and one or more network devices in communication with each other across a computer network. During normal operation of a computer cluster, the servers provide the network devices with computer resources and a place to store and retrieve data. In current computer cluster configurations the computer cluster data is stored on a shared computer disk that is accessed by any of the network servers.

A typical computer cluster is illustrated in Figure 1, which illustrates two network servers 110 and 120 in communication with network devices 130, 140, and 150 across computer network 101. Both network server 110 and network server 120 communicate with shared disk 104 across communication lines 105 and 106, respectively.

When using a computer cluster, it is often desirable to provide continuous availability of computer cluster resources, particularly where a computer cluster supports a number of user workstations, personal computers, or other network client devices. It is also often desirable to maintain uniform data between different file servers attached to a computer .clustering system and maintain continuous availability of this data to client devices. To achieve reliable availability of computer cluster resources and data it is necessary for the computer cluster to be tolerant of software and hardware problems or faults. Having redundant computers and a mass storage device generally does this, such that a backup computer or disk drive is immediately available to take over in the event of a fault.

A technique currently used for implementing reliable availability of computer cluster resources and data using a shared disk configuration as shown in Figure 1 involves the concept of quorum, which relates to a state in which one network server controls a specified minimum number of network devices such that the network server has the right to control the availability of computer resources and data in the event of a disruption of service from any other network server. The manner in which a particular network server obtains quorum can be conveniently described in terms of each server and other network devices casting "votes". For instance, in the two server computer cluster configuration of Figure 1, network server 110 and network server 120 each casts one vote to determine which network server has quorum. If neither network server obtains a majority of the votes, shared disk 104 then casts a vote such that one of the two network servers 110 and 120 obtains a majority, with the result that quorum is obtained by one of the network servers in a mutually understood and acceptable manner. Only one network server has quorum at any time, which ensures that only one network server will assume control of the entire network if communication between the network servers 110 and 120 is lost.

The use of quorum to attempt to make network servers available in the event of a disruption will now be described. There are two general reasons for which server 110 can detect a loss of communication with server 120. The first is an event, such as a crash, at server 120, in which server 120 is no longer capable of providing network resources to clients. The second is a disruption in the communication infrastructure of network 101 between the two servers, with server 120 continuing to be capable of operating within the network. If server 110 can no longer communicate with server 120, its initial operation is to determine if it has quorum. If server 110 determines that it does not have quorum, it then attempts to get quorum by sending a command to shared disk 104 requesting the disk to cast a vote. If shared disk 104 does not vote for server 110, this server shuts itself down to avoid operating independently of server 120. In this case, server 110 assumes that network server 120 is operating with quorum and server 120 continues to control the computer cluster. However, if shared disk 104 votes

for network server 110, this server takes quorum and control of the computer cluster and continues operation under the assumption that network server 120 has malfunctioned.

While the use of quorum to enable one of a plurality of network servers to continue providing network resources in the event of a disruption in the network is often satisfactory, the use of a shared disk places the entire network and the data stored on the disk at risk of being lost. For instance, if the shared disk 104, rather than one of the network servers 110 and 120 malfunctions, neither of the servers can operate, and the data may be permanently lost. Moreover, in a shared disk configuration the computer cluster servers are typically placed in close proximity to each other. This creates the possibility that natural disasters or power failures may take down the whole computer cluster.

SUMMARY OF THE INVENTION

The present invention relates to a method for improving the availability and reliability of computer cluster resources and data in a computer clustering system. Two servers each having an associated disk communicate across a computer network. Each server is capable of providing computer cluster resources and accessing computer cluster data for all network devices attached to the computer network. In the event of loss of communication, each server has the ability to determine the reason for loss of communication and determine whether or not it should continue operation.

When a network server detects that communication with another network server is lost, the loss in communication can be due to either a failure of the communication link or a failure of the other network server. Because each network server has a mirrored copy of the network data, a loss in communication is followed by execution of a series of acts at each network server that remains operating to ensure that the network servers do not begin operating independently of each other. In the absence of these acts, multiple network servers operating independently of one another could exist in an undesirable "split brain" mode, in which data mirroring between the network servers is not performed, thereby resulting in potential data corruption.

When operation of the computer cluster is initiated, one server is assigned control of the computer cluster resources and data and is given a "right to survive" in the event that communication between the network servers is lost as a result in failure of the communication link. For convenience, the one network server that has the "right to survive" during normal operation is designated herein as a "primary" server and any server that is not does not have the right to survive during normal operation is designated as a "secondary" server. It is noted that the terms "primary" and "secondary" do not connote relative importance of the servers, nor do they refer to which server is primarily responsible for providing network resources to network devices. Under normal operation, primary and secondary servers can be interchangeable from the standpoint of providing network resources. The right to survive is used in a default protocol to ensure that the split brain problem does not arise if communication between network servers is lost.

When a primary network server detects loss of communication, the primary network server can continue operating, since it can assume that the other, secondary network server has failed or that the secondary network server will not continue operation. The series of acts performed by a secondary network server upon detecting loss of communication is somewhat more complex. Rather than simply ceasing operation, the secondary network server infers or determines whether the loss of communication is a result of a failure of the primary network server or a failure in the communication link. If the communication link is operational, the secondary network server concludes that the primary network server has failed and is not operating. In this case, the secondary network server continues operating substantially without the risk of causing the split brain problem. If, however, the secondary network server determines that communication link has failed, it assumes that the primary network server is operational. In response to this assumption, the secondary network server terminates operation to avoid operating in a split brain mode.

A significant benefit according to the invention is that a secondary server, which does not initially have right to survive, can continue operating if it

determines that a loss of communication with the primary server is not caused by failure of the communication link. In the absence of any analysis of the communication link, the secondary server would be required to automatically shut down in response to a loss of communication with the primary server to avoid the split brain problem. It is noted that the foregoing methods of the invention for responding to loss of communication between servers enhances the reliability and availability of computer clusters in which each network server has a dedicated, mirrored disk or mass storage device, since the possibility of operating in a split brain mode does not force a secondary server to go off-line when a primary server fails.

Application of conventional "quorum" rules to computer clusters in which each network server has a dedicated, mirrored disk, is generally not optimal. For instance, in the case where a network server having quorum fails, there is no shared disk to cast a vote that would reassign quorum to the other network server. As a result, the direct application of conventional quorum rules to this type of computer cluster would result in the non-quorum network server unnecessarily shutting down upon failure of the network server having quorum.

Storing data in separate, mirrored disks greatly reduces the possibility of losing network data, which has been a problem frequently experienced in computer clusters having a single disk that is shared among network servers. Additionally, since servers do not share a single disk according to the invention, the location of the servers is not limited by the cable lengths associated with disk drive devices. Thus, network servers and their associated mirrored disks can be located remotely one from another. This reduces the chance that natural disasters or power failures may disable the entire computer cluster.

Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the

following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the above-recited and other advantages and features of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 is a block diagram illustrating a conventional computer cluster having servers that share a disk.

Figure 2 illustrates an exemplary computer cluster that provides a suitable operating environment for the present invention.

Figure 3 illustrates communication between the different software modules in a server to enable the server to decide whether to assume right to survive.

Figure 4 is a flow diagram illustrating a method whereby a server determines whether it is to assume the right to survive upon detecting loss of communication with the other server.

Figure 5 is a flow diagram illustrating a method whereby a server having the right to survive responds to a failure in another server in a computer cluster.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a method for improving the availability and reliability of computer cluster resources and data in a computer clustering system. The computer cluster includes at least two servers, each having a dedicated mass storage devices. The servers communicate with each other as well as other network devices across a computer network. Data is mirrored between the disks of each server so that network devices have access to reliable data in the event of one server failing. Communication modules operate on each server to

determine whether a server should shut itself down in the event of a communication loss between the servers.

The term "right to survive" refers to whether or not a server has the right to continue operation in the event of a detected loss of communication between the servers due to an error in the communication link between the servers. For example, if loss of communication between the two servers is detected, and such loss is due to a physical break in the communication link, the server with the right to survive continues operation while the server without right to survive shuts itself down. As noted previously, a "primary" server is one that has the right to survive prior to loss of communication, whereas a "secondary" server is one that does not have the right to survive prior to the loss of communication between servers.

The term "split brain" refers to an undesirable condition in which the network servers of a computer cluster having dedicated mass storage devices for each network server act independently of each other and without mirroring. In the absence of the invention, this can occur when communication between the network servers is lost while both network servers are still running, and each node assumes that the other has failed. When split brain occurs, the mirrored data on each server no longer matches and can be corrupt. Referring to the above example where the servers lose communication due to a physical break in the communication link, if the servers could not determine the loss was due the physical break, each would continue to operate and write its own data to its associated mass storage device. However, since the communication link is broken, the data would not be mirrored and inconsistent data on the mass storage devices would result. The present invention prevents the computer cluster from operating in a split brain mode, while enabling the secondary server to continue operating if the loss of communication is caused by failure of the primary server.

Embodiments within the scope of the present invention also include computer-readable media for carrying or having stored thereon computer-executable instructions or data structures. Such computer-readable media can be any available media, which can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-

readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a computer-readable medium. Thus, any such a connection is properly termed a computer-readable medium. Combinations of the above should also be included within the scope of computer-readable media. Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions.

Figure 2 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by computers in network environments. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represent examples of corresponding acts for implementing the functions described in such steps.

Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also

be practiced in distributed computing environments where tasks are performed by local processing devices and remote processing devices that are linked (either by hardwired links, wireless links, or by a combination of hardwired or wireless links) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Figure 2 illustrates a representative computer cluster configuration in which the method of the invention can be practiced. The computer cluster includes two servers, which are designated as server A 210 and server B 220. Although only two servers are illustrated in Figure 2, the general principles disclosed herein can be readily adapted to computer clusters having more than two network servers. Server A 210 and server B 220 both run a file operating system, which may be Microsoft Windows NT, although any of a variety of operating systems can be used with the invention. Server A 210 includes computer 212 connected to network 201 through network interface 211 and mass storage device 214 connected through mass storage controller 213. Likewise, server B 220 includes computer 222 connected to network 201 through network interface 220 and mass storage device 224 connected through mass storage controller 223. Network 201 can be an Ethernet, token ring, Arcnet, or any other network by which server A 210 and server B 220 can communicate with network devices 230, 240, and 250.

While it is not necessary for server A 210 to have identical components to server B 220, in many instances this will be the case. In other instances, server A 210 and server B 220 may have different processor types, different processor speeds, different size mass storage devices or any other number of hardware differences. All that is required of server A 210 and server B 220 is that they be capable of running the file operating system and that the drive on one of the servers not be so large that it cannot be mirrored to the other server.

In the embodiment illustrated in Figure 2, communication between server A 210 and server B 220 is established using dedicated link 202. Computer 212 is connected with dedicated link 202 through communication attachment 215 and

computer 222 is connected with dedicated link 202 through communication attachment 225. Dedicated link 202 can be implemented using a variety of techniques, well known to those skilled in the art. In one embodiment, dedicated link 212 is a link that uses an Ethernet protocol. Alternatives include using the serial communications ports of computers 212 and 222 programmed to run at high speeds or the parallel interfaces of computers 212 and 222. According to another alternative, dedicated link 202 and communication attachments 215 and 225 are not present, with the communication between server A 210 and server B 220 being established by a virtual circuit or channel carried across network 201. The specific type of hardware used to implement dedicated link 202 is not important, provided data transfer rates are comparable to the data transfer rates on mass storage devices 214 and 224 so performance of the system is not limited.

Network devices 230, 240 and 250 connect to network 201 through network interfaces 231, 241 and 251 respectively. These are client devices that use the resources of computer systems 210 and 220 to access data stored on the mass storage devices 214 and 224. Network devices 230, 240 and 250 can be any devices capable of communicating across network 201.

During operation of the computer cluster, both server A 210 and server B 220 are capable of providing resources to any of network devices 230, 240, or 250. Furthermore, both server A 210 and server B 220 are capable of sensing errors in the integrity of dedicated link 202. When a network device in communication with server A 210 adds, changes or deletes data from mass storage device 214, the update is mirrored across dedicated link 202 to mass storage device 224. Likewise, when a network device in communication with server B 220 adds, changes or deletes data from mass storage device 224 the update is mirrored across dedicated link 202 to mass storage device 214. Since the data is mirrored across dedicated link 202 in the embodiment of Figure 2, it does not congest network 201.

A result of mirroring is that mass storage devices 214 and 224 contain identical copies of all the computer cluster data. Therefore, the computer cluster can continue to operate reliably if one of the mass storage devices malfunctions.

If network devices 230, 240 and 250 cannot access data on mass storage 214 as a result of a malfunction of mass storage device 214, they may access the data on mass storage device 224 and vice versa. Since server A 210 and server B 220 do not share a mass storage device, there is no single point of failure and they may be remotely located from each other. Remotely locating server A 210 and server B 220 reduces the chance of a power outage or natural disaster preventing access to both mass storage devices simultaneously.

When server A 210 and server B 220 are initially configured, one of the servers is assigned the right to survive by the operator. The right to survive in this instance is represented as a software flag and is set to either "on" or "off." In general, during normal operation of the computer cluster, the right to survive flag is set to "on" on one and only one of the servers. After the initial configuration, a server with its right to survive flag "off" may turn its right to survive flag "on" under certain network conditions, as will be disclosed in greater detail below.

If communication between the servers is lost, the right to survive is used in determining which server is to continue cluster operations and which server is to terminate cluster operations. There are two primary instances when loss of communication between server A 210 and server B 220 occurs. The first instance occurs when one of the servers ceases communicating across dedicated link 202 as a result, for example, of the server experiencing a software error or the server losing power. The second instance occurs when there is a failure in the integrity of dedicated link 202.

As stated above, server A 210 and server B 220 can distinguish between types of communication failure. In the following discussion, it is assumed that server B 220 is a primary server, meaning that it has right to survive during an initial period of normal operation of the computer cluster, while server A 210 is a secondary server that does not initially have right to survive. If the primary server B 220 detects a loss of communication from server A 210, server B 220 continues operating in view of its right to survive without regard to whether the loss of communication has been caused by a failure of communication link 202 or of server A 210.

When secondary server A 210 detects a loss of communication from server B 220, server A 210 discontinues operation unless it determines that it is safe for it to assume right to survive. If server A 210 determines that the loss of communication is due to a failure in communication link 202, server A 210 assumes that primary server B 220 is operational. Accordingly, because server A 210 does not have right to survive, it terminates its activity so as to avoid operating in a split brain mode. If, however, secondary server A 210 determines that communication link 202 has not failed, it assumes that the loss of communication is associated with primary server 220 B failing or otherwise no longer providing network services. In this case, secondary server A 210 assigns itself right to survive and continues operation without the risk of causing a split brain in the computer cluster. In the latter case, server A 210 can service requests for resources from network devices 230, 240, and 250 that would otherwise be made to the failed server B 220. In any of the foregoing situations, only one server continues to function after loss of communication is experienced between the servers, which prevents the problem of split brain from occurring.

Figure 3 shows the interaction of different software modules running on a representative computer system according to one embodiment of the invention. The systems and structures of Figure 3 represent one example of how server A 210 can determine whether it should continue functioning and assume the right to survive upon detecting loss of communication with the other server. Right to survive flag 306 is initially set, either automatically or by a system administrator when the operation of the computer cluster is initiated. As stated above, the server with its right to survive flag 306 set to "on" continues functioning in the event of loss of communication with another server, regardless of whether the cause is an error in the integrity of the dedicated link 202 or a failure of the other server. While the invention broadly disclosed herein extends to a primary server that has its right to survive flag 306 initially set in the "on" position and that subsequently continues operation after experiencing a loss of communication, Figure 3 is discussed in detail below in the context of a secondary server that loses communication while the right to survive flag 306 is set in the "off" position.

As server A 210 operates in the computer cluster, the server receives and transmits data between components of the computer cluster as shown in Figure 3. Data 308 is representative of a heartbeat signal of server B 220 communicated on dedicated link 202 to server A 210. As used herein, the term "heartbeat signal" extends to any signal or data having any format that can used by one server to determine whether another server is operational. Communication attachment driver 301 is a software module that controls communication attachment 215 and receives data 308. Likewise, data 309 is representative of data transmitted on network 201. Network interface driver 302 controls network interface 211 and receives data 309.

Server A 210 can use data 308 and the modules that process and analyze data 308 and dedicated link 210 to determine whether to continue or discontinue operation in response to a determination that communication with server B 220 has been lost at a moment at which right to survive flag 306 is set in the "off" position. Server A 210 includes a server communication detector that monitors communication with server B 220 to determine whether data 308 is being received as expected. Link state detector 304 is any hardware component, software component, or any combination thereof that is capable of determining if dedicated link 202 has failed when communication with server B 220 is lost. In many cases, server communication detector 303 and link state detector 304 can be separate modules, although the functionality of these two components can be combined in a single structure.

In this example, communication detector 303 and link state detector 304 monitor data 308 and the integrity of dedicated link 202. However, in the alternative embodiment in which communication with server B 220 is transmitted using network 201 and network interface driver 302 rather than using dedicated link 202, the server communication detector 303 and the link state detector 304 monitor data 309 and the integrity of network 201.

In either case, server communication detector 303 determines whether the heartbeat signal of server B 220 is detected. In the case of a loss of the heartbeat signal of server B 220, link state detector analyzes the integrity of the

communication link that had been carrying the heartbeat signal (e.g., dedicated link 202 or network 201). When loss of communication with server B 220 is experienced, server communication detector 303 and link state detector 304 notify communication manager 305 of this fact and whether it appears that the communication link that had been carrying the heartbeat signal has failed. Loss of communication manager 305 then determines whether to shut server A 210 down or to permit server A to continue operating based on the right to survive flag 306 and the information received from server communication detector 303 and from link state detector 304.

To summarize the rules applied by loss of communication manager 305, server A 210 is permitted to continue operating if right to survive flag 306 indicates that server A 210 has the right to survive. Assuming, however, that server A 210 is a secondary server that does not have the right to survive according to right to survive flag 306 at the time that communication with server B 220 was lost, loss of communication manager 305 discontinues the operation of server A 210 if it has been determined that the loss of communication was a result of failure of the communication link. If loss of communication has been caused by failure of the communication link, it is assumed that server B 220 is operational, in which case, server A 210 discontinues operation, thereby avoiding the possibility of operating the computer cluster in a split brain mode.

In contrast, if loss of communication manager 305 determines that server A 210 detects a loss of communication from server B 220 and the communication link has not failed, loss of communication manager 305 infers that server B 220 has malfunctioning. Under the assumption that server A 210 is a secondary server, the right to survive flag 306 is then set to the "on" position, indicating that server A 210 has assumed the right to survive. In this case, server A 210 "reforms" the computer cluster by beginning to service requests previously made to server B 220. Server A 210 can do so by sending commands to network interface driver 302, causing network interface driver 302 to service requests that network devices have sent to server B 220 on network 201.

Figure 4 is a flow diagram showing steps performed at a secondary server (i.e., a server that does not already have the right to survive) in a computer cluster according to one embodiment of the invention for increasing the availability of a network resources in the event of a failure of a primary server (i.e., a server initially having the right to survive). It is noted that a primary server generally does not need to perform the steps illustrated in Figure 4 upon losing communication with a secondary server, as the right to survive possessed by primary server enables it to continue operating regardless of the reason for the loss of communication, as will be described below in greater detail in reference to Figure 5.

Turning first to Figure 4, the secondary server A attempts to detect reliable communication with server B in decision block 401. If server A can detect reliable communication with server B, the method proceeds to step 402, in which server A waits a certain time interval and then repeats decision block 401.

If server A does not detect reliable communication with server B in decision block 401, server A checks the reliability of the communication link in decision block 403. If server A does not detect a reliable communication link in decision block 402 server A terminates cluster activity at step 405 so as to avoid operating in a split brain mode.

If, however, server A does detect a reliable communication link at decision block step 403, the method proceeds to decision block 407, in which it is determined whether server A has waited the required number of time intervals before it can assume that server B has failed. If, according to decision block 407, server A has not waited the required number of time intervals, the method advances to step 408, in which server A waits for one time interval. After waiting during the time interval of step 408, server A attempts to detect reliable communication once again with server B in decision block 409. If server A detects reliable communication with server B in step 409, the method returns to step 402. Repeatedly checking for reestablished communication according to the loop defined by steps 407, 408, and 409 before reforming the cluster prevents

short duration non-fatal software errors in server B from causing server A to begin operating in the place of server B.

If server A fails to detect reliable communication from server B after a required number of attempts, server A assigns itself the right to survive in step 411 then reforms the cluster in step 412. After server A reforms the cluster in step 412, it once again starts checking for reliable communication from server B at decision block 413. If server B recovers from a non-terminating software error it might resume transmitting a heartbeat signal and otherwise communicating with server A after the cluster has been reformed. So long as communication from server B is not detected, server A continues operating and periodically monitors for communication from server B in the loop illustrated by decision block 413 and step 414. If communication from server B is reestablished as detected at decision block 413, processing advances to step 415, in which server B is shut down in step 415 before any disk access is permitted, thereby avoiding operation in a split brain mode and the associated data inconsistencies on the mass storage devices of server A and server B.

A primary server having right to survive performs the steps of the method of Figure 5 to respond to a loss of communication from another server according to one embodiment of the invention. In decision block 501, the primary server B attempts to detect reliable communication with server A. If server B can detect reliable communication with server A, the method proceeds to step 502, in which server B waits a certain time interval and then repeats decision block 501.

If server B does not detect reliable communication with server A in decision block 501, the method proceeds to decision block 507, in which it is determined whether server B has waited the required number of time intervals before it can assume that server A has failed. If, according to decision block 507, server B has not waited the required number of time intervals, the method advances to step 508, in which server B waits for one time interval. After waiting during the time interval of step 508, server B attempts to detect reliable communication once again from server A in decision block 509. If server B detects reliable communication with server A in step 509, the method returns to

step 502. Repeatedly checking for reestablished communication using the loop defined by steps 507, 508, and 509 before reforming the cluster prevents short duration non-fatal software errors in server A from causing server B to begin operating in the place of server A.

If server B fails to detect reliable communication from server A after a required number of attempts, server B reforms the cluster in step 512. After server B reforms the cluster in step 512, it once again starts checking for reliable communication from server A at decision block 513. If server A recovers from a non-terminating software error it might resume transmitting a heartbeat signal and otherwise communicating with server B after the cluster has been reformed. So long as communication from server A is not detected, server B continues periodically monitoring for communication from server A in the loop illustrated by decision block 513 and step 514. If communication from server A is reestablished as detected at decision block 513, processing advances to step 516, in which server B reforms the cluster to accommodate the resumed operation of server A.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed is:

1. A method for improving the availability and reliability of a computer clustering system including a first server and a second server connected by a communication link, wherein said second server is assigned the right to survive in case of disruption in said computer clustering system, said method comprising the acts of:

said first server detecting loss of communication from said second server to said first server;

said first server analyzing the communication link to determine if the communication link is functioning properly;

said first server continuing operation and assuming the right to survive if the communication link is determined to be functioning properly; and

said first server discontinuing operation if the communication link is determined to be not functioning properly.

2. A method as recited in claim 1 wherein the act of detecting loss of communication comprises the act of said first server determining that a heartbeat signal generated by said second server cannot be detected on the communication link.

3. A method as recited in claim 2, wherein the communication link comprises a dedicated link that connects the first server and the second server.

4. A method as recited in claim 1 wherein the communication link includes a virtual channel established in a computer network, said computer network comprising one or more network devices interconnected to each other and interconnected to said first server and said second server.

5. A method as recited in claim 1 wherein:

each of said first server and said second server further comprises a file operating system and at least one attached mass storage device; and

each of said first server and said second server executes said file operating system to become capable of servicing network requests from other network devices, said network requests comprising requests to use the resources of said first server and said second server.

6. A method as recited in claim 5, further comprising the acts of:

mirroring data from the at least one attached storage device of said first server to the at least one attached storage device of said second server; and

mirroring data from the at least one attached storage device of said second server to the at least one attached storage device said first server.

7. A method as recited in claim 6 wherein the communication link is a dedicated communication link that interconnects only said first server and said second server, the act of detecting loss of communication comprising the acts of:

said first server attempting to communicate with said second server across the communication link; and

said first server analyzing the results of the attempted communication across said communication link to determine said first server cannot communicate with said second server.

8. A method as recited in claim 1 wherein the act of analyzing the communication link to determine if said communication link is functioning properly further comprises the act of said first server repeatedly checking the transmission characteristics of the communication link to determine if they are within specified parameters.

9. A method as recited in claim 1 wherein said first server determines said communication link is functioning properly, the method further comprising the act of said first server servicing network requests made by a network device to said second server.

10. A method as recited in claim 9, further comprising the acts of:

said first server determining that communication from said second server is restored; and

said second server discontinuing operation.

11. A method as recited in claim 1 wherein said first server determines said communication link is not functioning properly, the method further comprising the act of said second server continuing to operate and service network requests made to said first server after said first server discontinues operation.

12. A method for improving the availability and reliability of a computer clustering system including a first server and a second server interconnected by a communication link, each of said first server and said second network server including a file server operating system and at least one associated mass storage device such that each of said first server and second server can receive requests that result in data being written to or read from the associated at least one mass storage device, said second server being assigned right to survive in case of disruption in said computer clustering system, said method comprising the acts of:

said first server determining that a heartbeat signal associated with said second server is no longer being detected on said communication link;

said first server analyzing the communication link to determine if any error exists in the integrity of the communication link;

if said first server determines there is an error in the integrity of the communication link, said first server discontinuing operation and said second server reforming the computer clustering system so that said second server services requests that would otherwise be directed to said first server; and

if said first server determines that there is no error in the integrity of the communication link, said first server assigning itself the right to survive and said first server reconfiguring the computer clustering system so that said first server services requests that would otherwise be directed to said second server.

13. A method as recited in claim 12, further comprising, prior to the act of determining that the heartbeat signal associated with the second server is no longer being detected, the act of said first server and said second server mirroring data stored on the at least one mass storage device of the first server and the at least one mass storage device of the second server.

14. A method as recited in claim 13, wherein the act of mirroring data comprises transmitting the data on the communication link between the first server and the second server.

15. A method as recited in claim 13, wherein the act of said first server determining that the heartbeat signal associated with said second server is no longer being detected comprises the acts of:

said first server repeatedly monitoring the communication link for the heartbeat signal; and

said first server determining that the heartbeat signal is no longer being detected when said first server does not detect the heartbeat signal during the act of repeatedly monitoring the communication link during a specified period of time.

16. A method as recited in claim 12, further comprising, after the act of said first server reconfiguring the computer clustering system, the acts of:

said first server again detecting the heartbeat signal associated with the second server; and

said second server discontinuing operation prior to accessing the at least one mass storage device associated with said second server.

17. A method for improving the availability and reliability of a computer clustering system including a first server and a second server interconnected by a communication link, each of said first server and said second network server including a file server operating system and at least one associated mass storage device such that each of said first server and second server can receive requests that result in data being written to or read from the associated at least one mass storage device, said second server being assigned right to survive in case of disruption in said computer clustering system, said method comprising the acts of:

during normal operation of the computer clustering system, mirroring data on the at least one mass storage device associated with the first server and on the at least one mass storage device associated with the second server by transmitting the data between the first server and the second server using the communication link;

said first server determining that said second server is not functioning properly, including the acts of:

said first server analyzing the communication link at specified time intervals;

said first server failing to detect communication from said second server on said communication link; and

said first server detecting proper functionality of said communication link based on the act of analyzing said communication link;

said first server taking control of the computer clustering system; and

said first server reconfiguring the computer clustering system so said first server receives file server requests that would otherwise be directed to said second server.

18. A method as recited in claim 17, further comprising the acts of:

said first server determining that said second server has reestablished proper functionality after said first server has taken control of the computer clustering system; and

said second server discontinuing operation prior to accessing said at least one mass storage device associated with said second server.

19. A computer program product for implementing, in a first server included in a computer clustering system that also includes a second server and a communication link connecting the first server and the second server, a method for said first server to assume control of the computer clustering system in response to a failure of said second server, the computer program product comprising:

a computer-readable medium carrying computer-executable instructions for implementing the method, said computer-executable instructions including:

program code means for determining whether said first server has a right to survive if said first server fails to detect

communication from said second server, said first server initially not having the right to survive;

program code means for detecting loss of communication from said second server;

program code means for determining, in response to the loss of communication, whether the communication link is functioning properly;

program code means for executing the acts of:

if the communication link is not functioning properly, discontinuing operation of said first server; and

if the communication link is functioning properly, continuing operation of said first server, notwithstanding said first server initially not having the right to survive.

20. A computer program product as recited in claim 19, wherein the communication link comprises a dedicated link between the first server and the second server.

21. A computer program product as recited in claim 19, wherein the communication link comprises a virtual channel included in a network that interconnects the first server, the second server, and a plurality of network devices.

22. A computer program product as recited in claim 19, wherein the computer-executable instructions further comprise program code means for mirroring data between a mass storage device associated with the first server and a mass storage device associated with the second server during normal operation of the computer clustering system, while both the first server and the second server communicate one with another.

23. A computer program product as recited in claim 19, wherein the program code means for determining whether said first server has a right to survive comprises program code means for examining a right to survive flag stored at the first server.

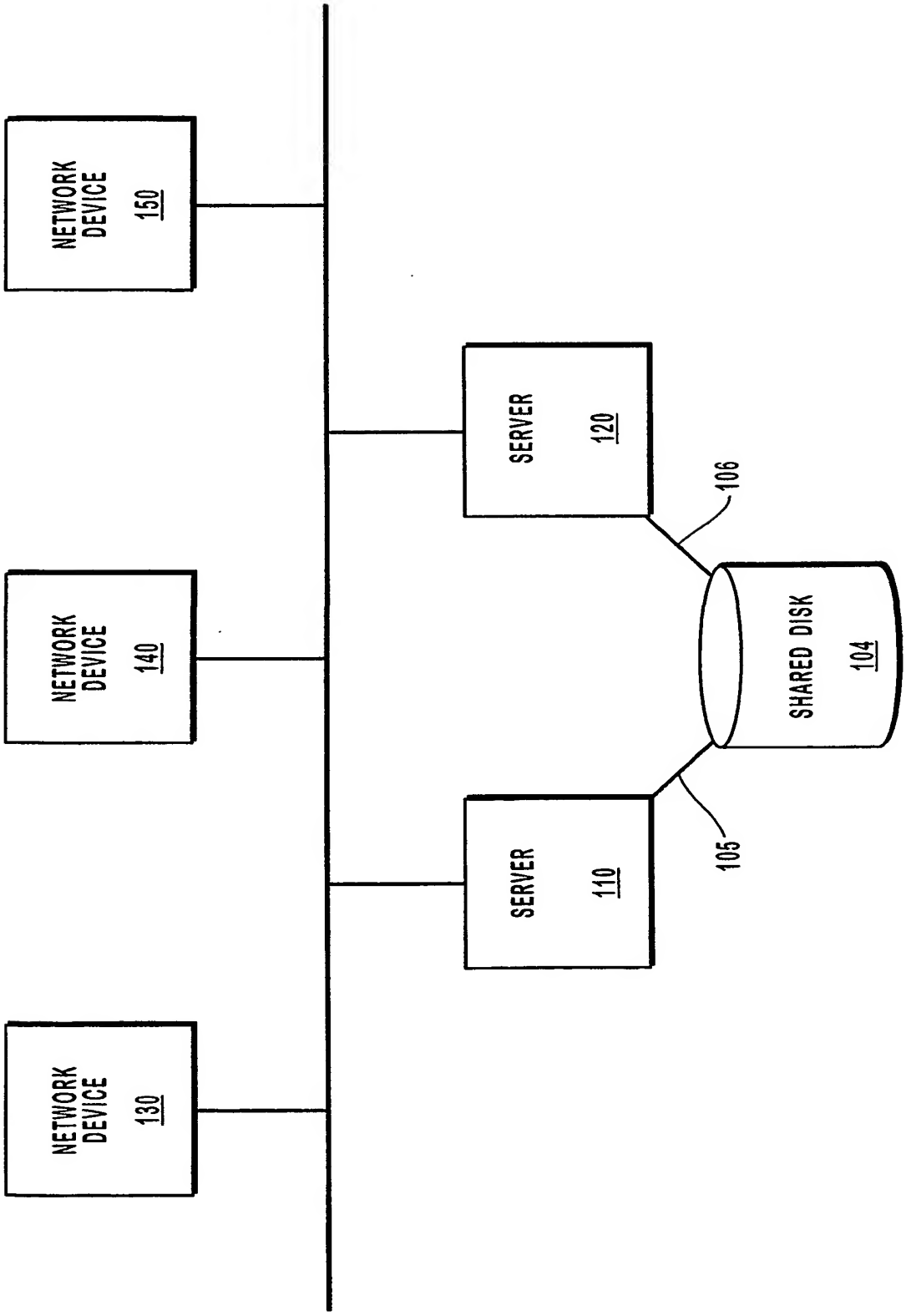


FIG. 1
(PRIOR ART)

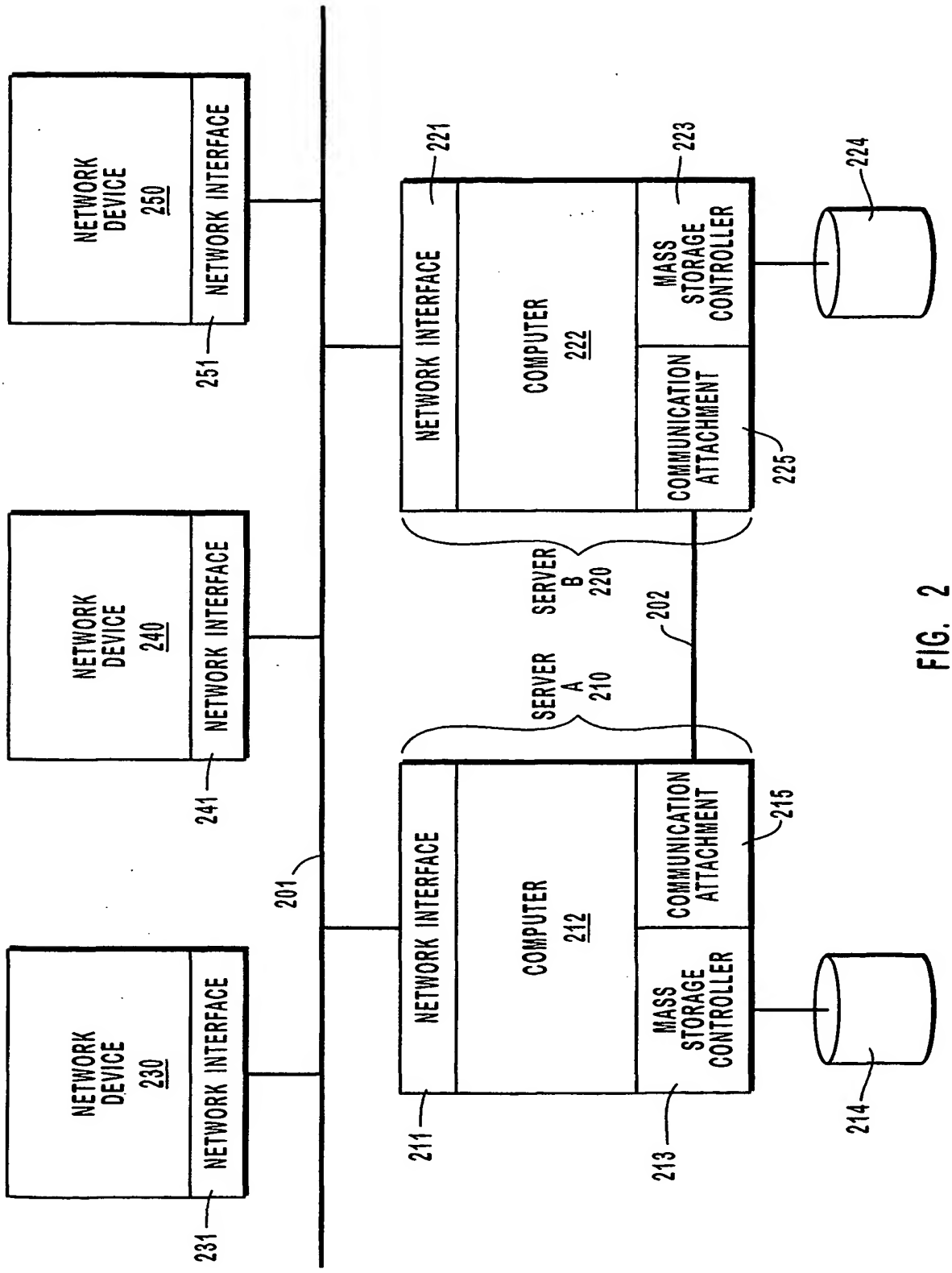

NETWORK DEVICE 250
NETWORK INTERFACE
251
NETWORK DEVICE 240
NETWORK INTERFACE
241
NETWORK DEVICE 230
NETWORK INTERFACE
231
201
NETWORK INTERFACE
COMPUTER 212
COMMUNICATION ATTACHMENT
MASS STORAGE CONTROLLER
211
213
215
214
SERVER A 210
202
NETWORK INTERFACE
COMPUTER 222
COMMUNICATION ATTACHMENT
MASS STORAGE CONTROLLER
221
223
225
224
SERVER B 220

FIG. 2

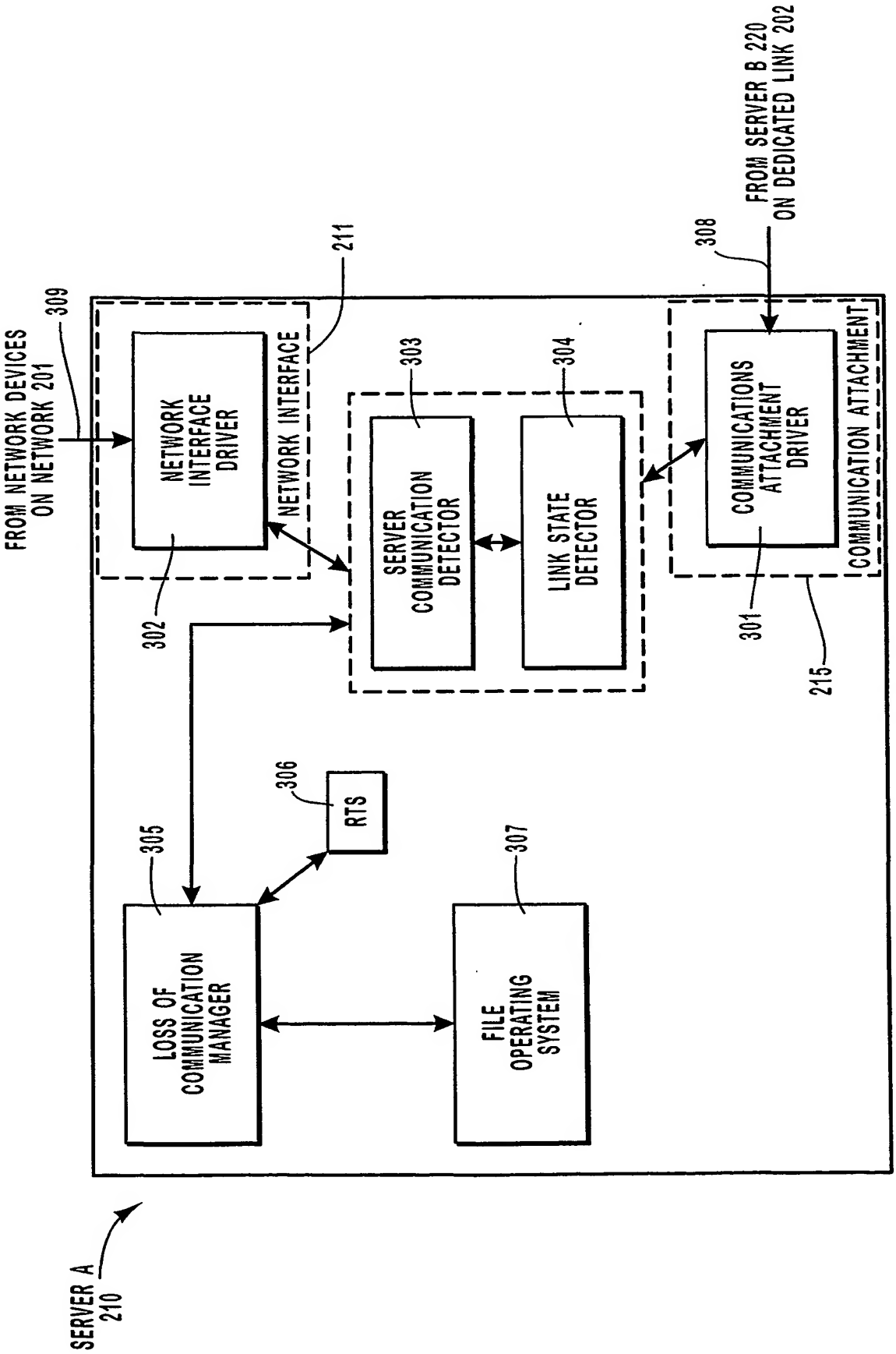

SERVER A
210
FROM NETWORK DEVICES
ON NETWORK 201
309
NETWORK
INTERFACE
DRIVER
NETWORK INTERFACE
211
302
303
SERVER
COMMUNICATION
DETECTOR
304
LINK STATE
DETECTOR
308
FROM SERVER B 220
ON DEDICATED LINK 202
COMMUNICATIONS
ATTACHMENT
DRIVER
COMMUNICATION ATTACHMENT
301
215
306
RTS
305
LOSS OF
COMMUNICATION
MANAGER
307
FILE
OPERATING
SYSTEM

FIG. 3

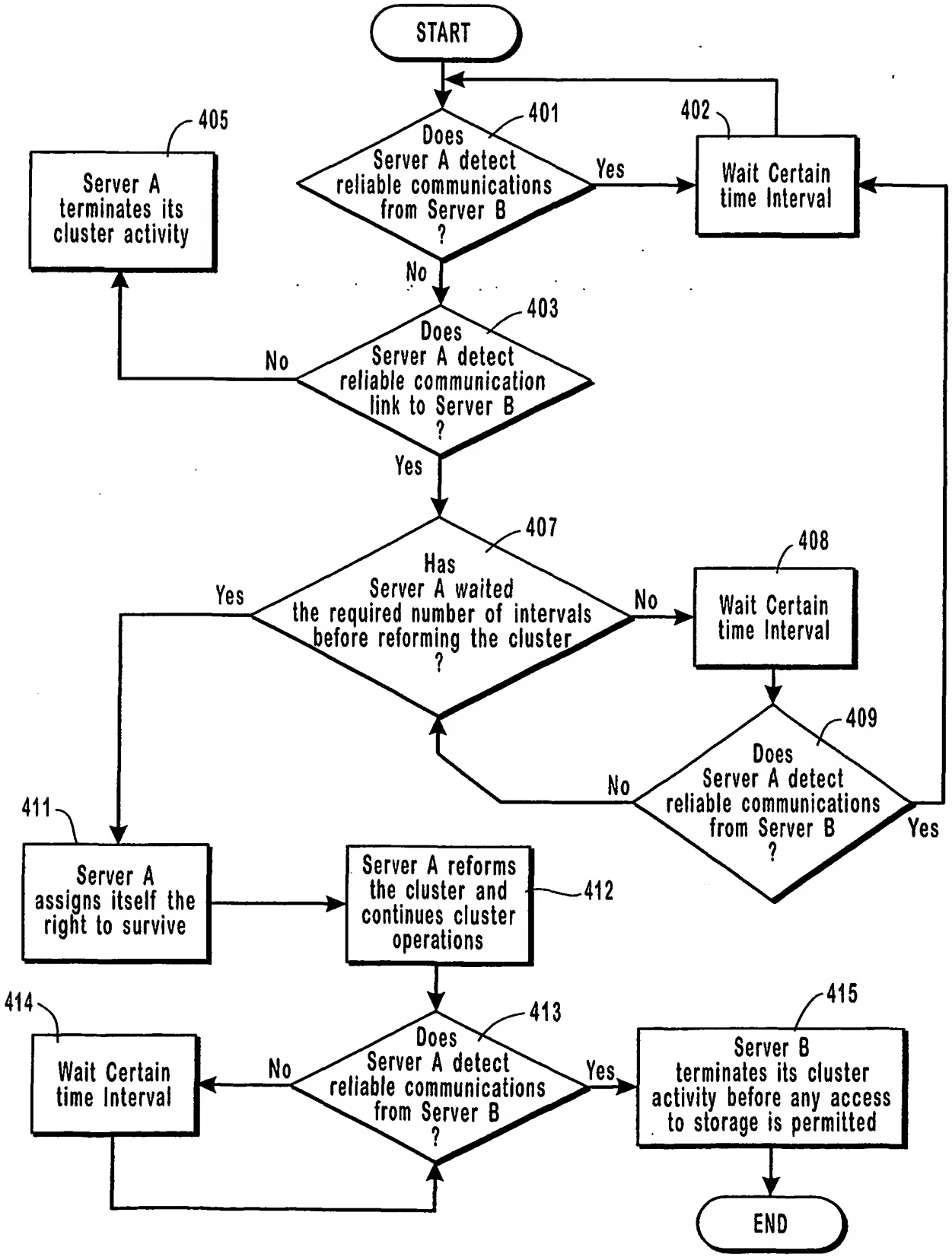

START
401
Does Server A detect reliable communications from Server B ?
Yes
402
Wait Certain time Interval
No
405
Server A terminates its cluster activity
403
Does Server A detect reliable communication link to Server B ?
No
Yes
407
Has Server A waited the required number of intervals before reforming the cluster ?
Yes
No
408
Wait Certain time Interval
409
Does Server A detect reliable communications from Server B ?
No
Yes
411
Server A assigns itself the right to survive
Server A reforms the cluster and continues cluster operations
412
414
Wait Certain time Interval
413
Does Server A detect reliable communications from Server B ?
No
Yes
415
Server B terminates its cluster activity before any access to storage is permitted
END

FIG. 4

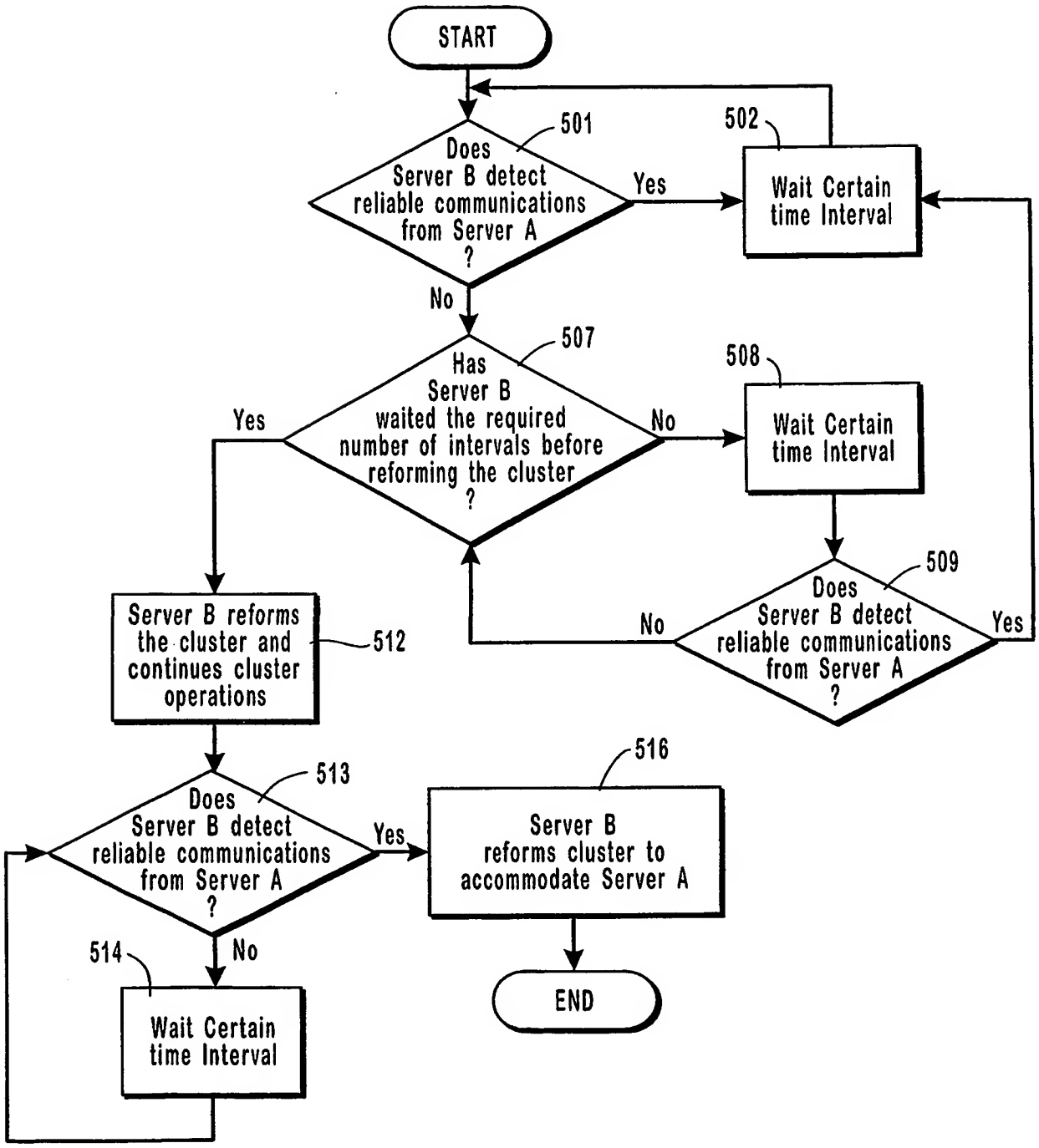

START
501
Does Server B detect reliable communications from Server A ?
Yes
502
Wait Certain time Interval
No
507
Has Server B waited the required number of intervals before reforming the cluster ?
Yes
No
508
Wait Certain time Interval
509
Does Server B detect reliable communications from Server A ?
No
Yes
Server B reforms the cluster and continues cluster operations
512
513
Does Server B detect reliable communications from Server A ?
Yes
516
Server B reforms cluster to accommodate Server A
No
514
Wait Certain time Interval
END

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/49600

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 11/00, 11/20, 11/30, 11/14, 11/16
US CL : 714/4
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 714/4

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
E,A	US 6,353,898 B1 (WIPFEL et al) 5 March 2002 (05.03.2002), entire document.	1-23
E,A	US 6,338,112 B1 (WIPFEL et al) 8 January 2002 (08.01.2002), entire document.	1-23
P,A	US 6,311,217 B1 (EHLINGER et al) 30 October 2001 (30.10.2001), entire document.	1-23
P,A	US 6,279,032 B1 (SHORT et al) 21 August 2001 (21.08.2001), entire document.	1-23
P,A	US 6,192,483 B1 (MOIIN et al) 20 February 2001 (20.02.2001), entire document.	1-23
A	US 6,145,089 A (LE et al) 7 November 2000 (07.11.2000), entire document.	1-23
A	US 6,108,699 A (MOIIN) 22 August 2000 (22.08.2000), entire document.	1-23
A	US 6,002,851 A (BASAVAIAH et al) 14 December 1999 (14.12.1999), entire document.	1-23
A	US 5,999,712 A (MOIIN et al) 7 December 1999 (07.12.1999), entire document.	1-23

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 March 2002 (08.03.2002)	12 APR 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer John E. Breene Peggy Harrod Telephone No. 703-305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/49600

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,991,518 A (JARDINE et al) 23 November 1999 (23.11.1999), entire document.	1-23
A	US 5,948,109 A (MOIIN et al) 7 September 1999 (07.09.1999), entire document.	1-23
A	US 5,892,895 A (BASAVAIAH et al) 6 April 1999 (06.04.1999), entire document.	1-23
A	US 5,884,018 A (JARDINE et al) 16 March 1999 (16.03.1999), entire document.	1-23
A	US 5,828,889 A (MOIIN et al) 27 October 1998 (27.10.1998), entire document.	1-23
A	WO 98/33121 A1 (TANDEM COMPUTERS INCORPORATED) 30 July 1998 (30.07.1998), entire document.	1-23
A	EP 0 810 526 A1 (SUN MICROSYSTEMS, INC.) 3 December 1997 (03.12.1997), entire document.	1-23
A	MURRAY, P.T. et al Somersault: Enabling Fault-Tolerant Distributed Software Systems, HP Laboratories Publication HPL-98-81, April 1998, entire document.	1-23
A	YANG, C-L et al Hybrid Fault Diagnosability with Unreliable Communication Links, IEEE TRansactions on Computers, Vol. 37, No. 2, February 1988, entire document.	1-23

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/49600

Continuation of Item 4 of the first sheet:
The proposed title fails to satisfy the requirements of PCT Rule 4.3, because it is not short and precise, and exceeds 17 words.

The proposed new title is "Method of 'Split-Brain' Prevention in Computer Cluster Systems".

Continuation of B. FIELDS SEARCHED Item 3:
IEEE Explore, ACM
search terms: split brain, cluster

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 June 2002 (27.06.2002)

PCT

(10) International Publication Number
WO 02/050678 A1

(51) International Patent Classification[7]: G06F 11/00, 11/20, 11/30, 11/14, 11/16

(21) International Application Number: PCT/US01/49600

(22) International Filing Date:
19 December 2001 (19.12.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/257,478 21 December 2000 (21.12.2000) US
09/855,592 14 May 2001 (14.05.2001) US

(71) Applicant: **LEGATO SYSTEMS, INC.** [US/US]; 2350 West El Camino Real, Mountain View, CA 94040 (US).

(72) Inventor: **PRICE, Daniel, M.**; 11030 Manor Circle, Highland, UT 84003 (US).

(74) Agents: **ISRAELSEN, R., Burns** et al.; Workman, Nydegger & Seeley, 1000 Eagle Gate Tower, 60 East South Temple, Salt Lake City, UT 84111 (US).

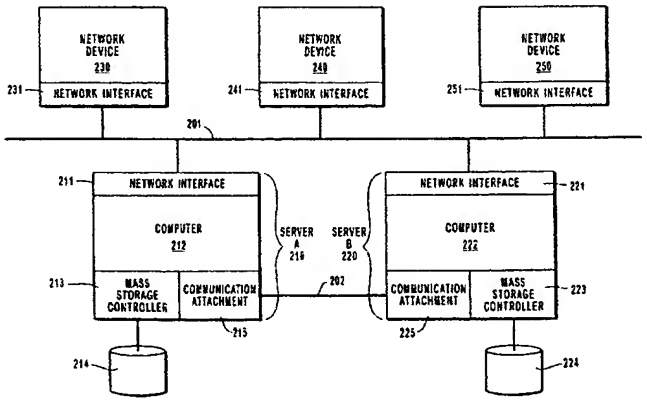
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: METHOD OF "SPLIT-BRAIN" PREVENTION IN COMPUTER CLUSTER SYSTEMS



(57) Abstract: A method for increasing the availability of a first server (210) included in a computer cluster when a second server (220) fails. Each server (210, 220) in the computer cluster has an associated mass storage device 214, 224 and can process requests from any network device (230, 240, 250) in the computer cluster. Data is mirrored between the mass storage devices (214, 224) of the servers (210, 220) so that each server's mass storage device has a complete copy of all computer cluster data. Data mirroring takes place across a dedicated link (202), which reduces congestion on the rest of the computer cluster. When the first server (210) detects a loss of communication from the second server (220), the first server (210) determines if the loss of communication is a result of a malfunction of the dedicated link (202). If the dedicated link (202) has failed, the first server (210) discontinues operation to avoid writing data to its associated mass storage device (214), which cannot be mirrored due to the loss of communication. If the dedicated link (202) is operational, the first server (210) continues operation. In either case, since each server (210, 220) can process requests from any network device (230, 240, 250) and each server has a complete copy of all the network data, the computer cluster continues to be available for use even after a server is shut down.

WO 02/050678 A1

(48) Date of publication of this corrected version:
19 September 2002

(15) Information about Correction:
see PCT Gazette No. 38/2002 of 19 September 2002, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.